

基于集成学习的极化码BP译码算法

徐 坤

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2023年3月23日; 录用日期: 2023年4月20日; 发布日期: 2023年4月29日

摘 要

集成学习被广泛用于解决复杂任务, 方法是通过学习得到不同的弱学习器, 利用其各自的泛化能力, 获得强学习器来解决问题。BPL译码算法和WBP译码算法是BP译码算法的两种主要优化算法, 两种优化算法均具有不同的优势。基于此, 提出了一种基于集成学习的BP译码算法, 该算法借助集成学习的优势, 来获取更优异的译码性能。通过训练多组具有不同的权重参数的WBP译码器, 并将其组合成类似BPL译码算法的一种新的译码器, 该译码器的权重参数由离线训练得来。仿真结果表明, 该译码算法可以达到与BPL译码算法相近的译码性能, 同时具有更低的计算复杂度及更少的迭代次数。

关键词

极化码, BP译码, 加权BP译码, 集成学习, 弱学习器

BP Decoding Algorithm Based on Ensemble Learning

Kun Xu

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology
Shanghai

Received: Mar. 23rd, 2023; accepted: Apr. 20th, 2023; published: Apr. 29th, 2023

Abstract

Ensemble learning is widely used to solve complex tasks by learning different individual learners, using their respective generalization capabilities to obtain strong learners to solve problems. BPL decoding algorithm and WBP decoding algorithm are the two main optimization algorithms of BP decoding algorithm, and both optimization algorithms have different advantages. Based on this, a BP decoding algorithm based on ensemble learning is proposed, which obtains better decoding performance by virtue of the advantages of ensemble learning. By training multiple sets of WBP decoders with different weight parameters, and combining them into a new decoder similar to the

BPL decoding algorithm, the weight parameters of the decoder are obtained from offline training. Simulation results show that the decoding algorithm can achieve similar decoding performance as the BPL decoding algorithm, and has lower computational complexity and fewer iterations.

Keywords

Polar Code, BP Decoding, Weighted BP Decoding, Integrated Learning, Individual Learner

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

由于信道噪声的存在, 信道编码是无线通信至关重要的研究领域, 是保障信息可靠、准确、有效传输的关键。极化码是信道编码领域的新突破, 是唯一一种在二进制离散无记忆信道中能够达到信道香农限的编码方案[1]。与 LDPC 码和 Turbo 码相比, 极化码具有较好的译码性能和低复杂度。同时, 极化码也被广泛应用于存储系统、深空通信、水下通信、光通信等场景[2]。

BP 译码算法是一种并行迭代译码方案, 具有译码延迟低、吞吐率高的特点。但在译码性能上, 与 SCL 译码相比, 存在一定差距。在低信噪比通信环境下需要较多次迭代才能成功译码, 其性能很大程度上取决于迭代次数。研究者设计多种早停机机制来提前结束译码, 降低迭代次数来提高收敛速度[3]。为进一步提高 BP 译码算法性能, Elkelesh 等人[4]提出了一种置信传播列表(BPL)译码器, 其性能与极化码的连续取消列表(SCL)解码器相当。BPL 译码器由多个基于不同排列极化码因子图的并行独立置信传播(BP)解码器组成。为了降低 BPL 译码器带来的空间复杂度, Doan 等人利用比特索引来调度极化码因子图[5] [6], 使得串行 BPL 译码器可以复用同一个 BP 译码器。有研究者[7]提出了一种依赖于人工噪声的新型信念传播列表(BPL)译码器, 该解码器将不同功率的噪声添加到并行独立的 BP 译码器中解决译码过程中未收敛的问题。Geiselhart 等人[8]利用循环冗余校验(CRC)作为停止条件, 提出的 CRC 辅助信念传播列表(CA-BPL)译码算法, 该算法译码性能接近 CA-SCL。Ren 等人[9]基于环路简化(Loop Simplification, LS)对极化码因子图进行筛选, 所提出的 LS-BPL 译码算法相比于 BPL 译码算法具有更好的译码性能和更低的平均解码延迟。置信传播翻转(BPF)译码[10]算法不同于其他 BP 的优化算法, 它利用解码过程中的信息, 更有针对性地翻转易错比特。为了更有效地识别易错比特, 广义 BPF (GBPF)译码算法和增强 BPF (EBPF)译码算法[11] [12]用来提高翻转集合识别错误比特的准确率和实现多比特翻转。

近年来, 基于机器学习的极性 BP 解码器成为了热门的研究方向。Doan 等人[13]将极化码因子图排列选择问题看作强化学习中的多臂老虎机问题, 通过在线学习方式在 BPL 译码过程中选择最优的极化码因子图。同时, 由于神经网络具有并行结构, 可以泛化 BP 解码结构, 常用于与 BP 译码算法相匹配来提高译码吞吐率和降低时延[14] [15]。Nachmani 等人[16] [17]提出加权置信传播(WBP)译码算法, 该算法使用深度神经网络模型训练加权 BP 译码算法。但当码字长度增加时, 训练复杂度呈指数级增长。Lugosch 等人[18]使用偏移最小和算法来代替求和 - 乘积算法, 降低训练过程中的计算复杂度。Xu 等人[19]基于 WBP 译码算法提出了一种适应各种码长的多尺度置信传播(BP)算法。Teng 等人[20]基于码本对递归神经网络权重进行量化, 采用了迭代架构, 使各种迭代间可共享训练的权重。Huang 等人[21]提出了基于残差神经网络的 BP 解码器(ResNet-BP), 以补充传统的 BP 算法, 该算法具有较低复杂度的优势。Oliveira

等人[22]等人利用强化学习来优化 WBP 译码算法的加权参数。Raviv 等人[23] [24]最先将数据驱动集成方案应用于极化码,用于集成 BPL 译码和 WBP 译码,每个译码器都被训练来解码不同的数据,同时将 CRC 码作为门控器对码字进行分类和聚合。

BPL 译码器提供了广度的译码范围,而 WBP 译码器则有着优于 BP 译码器的优势,所以两者集成是一种新颖的学习策略。本文基于两者优势,提出了一种基于集成学习的 BP 译码算法,通过分析 BPL 译码器与集成学习的结构特点,将 WBP 译码器作为弱学习器对其权重参数进行单独离线训练,并利用学习器筛选来组合强学习器用于信道译码,该算法可以有效地提升 BP 译码算法的性能。首先,将每个 WBP 译码器看作弱学习器,利用不同的数据集来分类训练,训练后的 WBP 译码器具有不同的译码能力,WBP 译码器的训练可以离线完成。然后,再由学习器筛选组合一个强学习器(E-WBP 译码器),该学习器有若干个弱学习器组合而成。最后,译码时只需要将训练权重参数送至 E-WBP 译码器中,对接收到的信源序列进行译码,由早停机制结束译码。

2. BP 译码与 BPL 译码算法

2.1. BP 译码算法

置信传播译码(Belief Propagation, BP)算法是一种基于极化码因子图的迭代算法,广泛用于 Turbo 码、LDPC 码等。BP 译码算法具有并行译码、高吞吐率的特点,硬件实现更容易。定义极化码 $P(N, R, \mathcal{A})$ 的码长 N , 码率 $R = K/N$, 信息位长度为 K , 信息比特集合 \mathcal{A} 。其极化码因子图有 n 层, $n = \log_2 N$, 共包含 $(n+1) \times N$ 个节点, (i, j) 表示第 i 节第 j 级的节点, 其中 $1 \leq i \leq n+1$, $1 \leq j \leq N$ 。令 $L_{i,j}$ 和 $R_{i,j}$ 分别表示节点 (i, j) 的左向传播和右向传播, 每次左向传播和右向传播都会对节点信息更新如图 1 所示。

$$\begin{cases} L_{i,j} = f(L_{i+1,j}, R_{i,j+2^{n-1-i}} + L_{i+1,j+2^{n-1-i}}) \\ L_{i,j+2^{n-1-i}} = f(L_{i+1,j}, R_{i,j}) + L_{i+1,j+2^{n-1-i}} \\ R_{i+1,j} = f(R_{i,j}, R_{i+1,j+2^{n-1-i}} + L_{i+1,j+2^{n-1-i}}) \\ R_{i+1,j+2^{n-1-i}} = f(R_{i,j}, L_{i+1,j}) + R_{i,j+2^{n-1-i}} \end{cases} \quad (1)$$

其中, $f(\alpha, \beta) \triangleq \ln \frac{1 + e^{\alpha + \beta}}{e^\alpha + e^\beta}$, 式(1)已由最小和译码算法式(2)简化:

$$f(\alpha, \beta) \approx \text{sign}(\alpha, \beta) \min(|\alpha|, |\beta|) \quad (2)$$

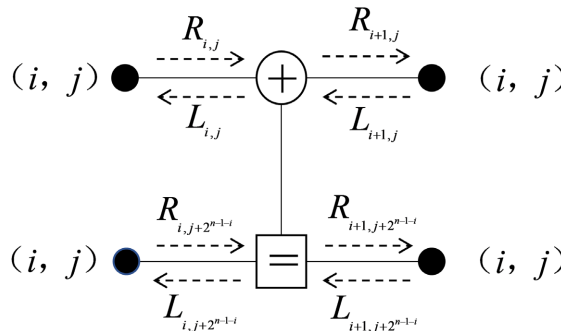


Figure 1. Polarization code BP decoding algorithm minimum computing unit

图 1. 极化码 BP 译码算法最小计算单元

在极化码的 BP 译码前, BP 译码算法需要 $L_{i,j}$ 根据信道接收的后验 LLR 值初始化, $R_{i,j}$ 初始化是直

接将冻结比特对应的 $R_{i,j}$, $j \in \mathcal{A}^c$ 设置为无穷大, 否则设置为 0。当 BP 译码算法达到最大迭代次数 T_{\max} 或满足其他早停条件时停止译码, 通过硬判决公式计算出译码结果:

$$\hat{u}_j = \begin{cases} 0, & \text{if } R_{1,j} + L_{1,j} \geq 0 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

2.2. BPL 译码算法

BP 译码算法最小计算单元可以组成不同的极化码因子图, BP 译码时使用不同的因子图会有不同的译码性能, 这也是 BPL 译码算法增益的主要原因。码长为 N 的极化码可以得到最多 $(\log_2 N)!$ 个因子图, 对于 BPL 译码算法的难题就是从 $(\log_2 N)!$ 个因子图中筛选出 L 个译码性能较优的因子图组成最后的译码方案。图 2 中图 2(a), 图 2(b) 展示了 $N=8$ 时, 两种不同排列类型的因子图。

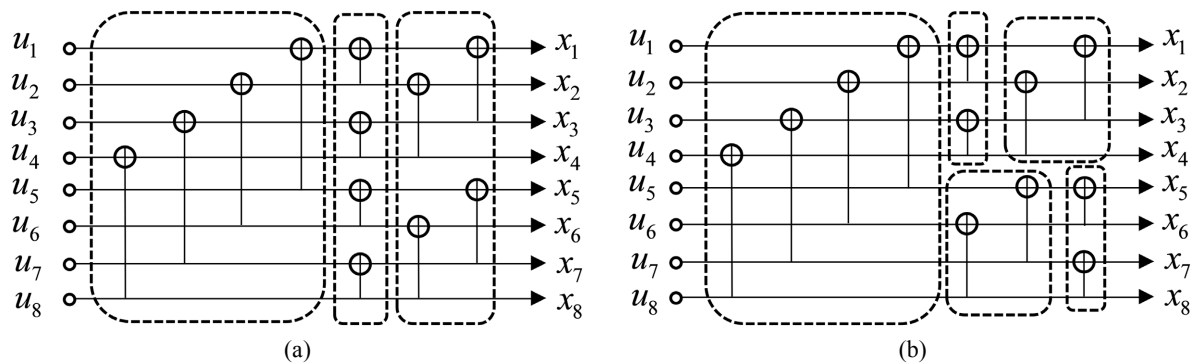


Figure 2. The arrangement of the two factor plots of polarization codes with a length of 8
图 2. 码长为 8 的极化码两种因子图排列方式

由于因子图与码字位置存在映射关系, 可以利用置换比特索引的位置来表示不同的因子图, 使得 BPL 译码算法可以复用同一个因子图。如果将因子图中的“XOR”运算看作 1, “dot”运算看作 0, 可以将因子图的排序用矩阵向量的方式展示出来如, 可以通过线性变换来实现矩阵向量的转换, 这种线性变换也就是置换比特索引的表示形式。如图 3 所示, 分别是图 2(a)和图 2(b)对应的因子图的矩阵形式, 可以看作因子图与码字索引对应的关系, 而且两者之间可以通过置换比特索引来改变索引排序, 从而实现 BP 译码器的复用。

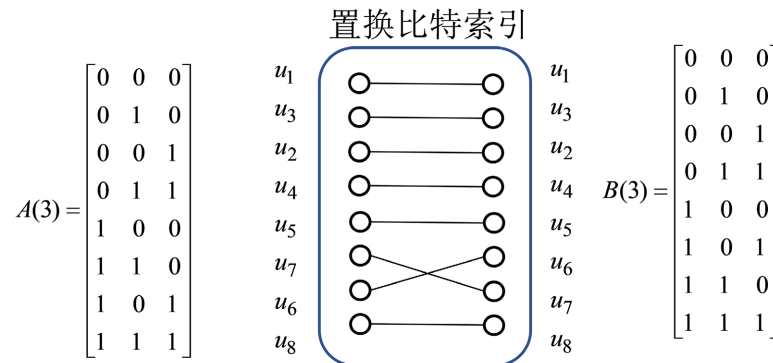


Figure 3. The correspondence between different factor plots of polarization codes with a length of 8
图 3. 码长为 8 的极化码不同因子图间对应关系

译码器接收到 y_1^N 时, 会并行送到 L 个译码器, 每个译码器因子图为 F_i ($1 \leq i \leq L$)。所有的因子图将会获得相同的信号 y_1^N , 每个独立运行的译码器 $BP(F_i)$ 都将经过多次迭代, 输出相应的信源序列估计

$\hat{u}_i (1 \leq i \leq L)$ 和码字估计 $\hat{x}_i (1 \leq i \leq L)$ ，然后根据设置的早停机制可以提前终止 $BP(F_i)$ 的译码。通常使用基于生成矩阵早停机制来判断编码约束关系 $\hat{u}_i G_N = \hat{x}_i$ ，判决是否结束 $BP(F_i)$ 译码。最后再利用最小距离译码准则确定出最后的译码码字 \hat{x}_{BPL} 。每个译码器都是具有不同结构排列的因子图，可以利用因子图与码字间的映射关系来使 BP 译码器可以复用，降低硬件实现难度。基于生成矩阵早停机制的极化码 BPL 译码算法流程图如图 4 所示。同时还有 CRC 早停机制，CRC 早停机制就是在发送端对码字进行 CRC 编码，在迭代的译码周期中对估计码字 $\hat{u}_i (1 \leq i \leq L)$ 进行 CRC 检测，若通过则停止译输出最后的译码码字 \hat{x}_{BPL} 。

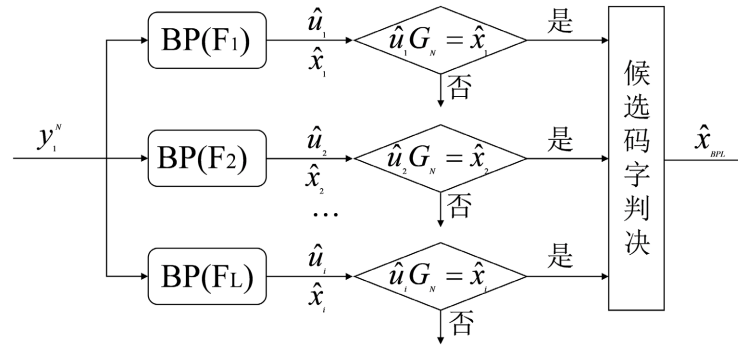


Figure 4. Flow chart of polarized code BPL decoding algorithm based on the generation matrix early stop mechanism
图 4. 基于生成矩阵早停机制的极化码 BPL 译码算法流程图

候选码字的判决，采用 BPL 译码器最小距离译码准则：

$$\hat{x}_{BPL} = \arg \min_{\hat{x}_i} \|y_i^N - \hat{x}_i\| \quad (4)$$

3. 基于集成学习的极化码 BP 译码算法

集成学习在分类、聚类、半监督学习等方面表现出良好的准确性和稳定性，在很多不同的领域，如数据挖掘，文本分类，生物信息学和人脸图像分类等得到了应用，受到越来越多的关注。集成学习的主要思想是将多个弱学习器(弱学习器)结合，构建成一个有较强性能的机器学习器。

3.1. WBP 译码算法

近几年机器学习在很多领域都被广泛使用，同时机器学习(深度学习)方法也被应用于通信的信道译码算法中，特别是与 BP 译码算法结合。神经网络由多层连接的神经元组成，包含输入层、隐藏层、输出层。基于因子图表示的 BP 译码算法经过多次迭代所形成的结构，可以看作是一种有固定计算方式的特殊神经网络。加权 BP(WBP)译码算法利用因子图的迭代结构来实现，将因子图里的最小计算单元参数化。设置迭代过程中最小计算单元的每个节点都有两个额外的参数 $r_{i,j}^{L,t}$ 和 $r_{i,j}^{R,t}$ 分别表示在第 t 次迭代时因子图中 (i, j) 节点左消息和右消息的可训练权重。

$$\begin{cases} L_{i,j}^t = r_{i,j}^{L,t} f(L_{i+1,j}^t, R_{i,j+2^{n-1-i}}^t + L_{i+1,j+2^{n-1-i}}^t) \\ L_{i,j+2^{n-1-i}}^t = r_{i,j+2^{n-1-i}}^{L,t} f(L_{i+1,j}^t, R_{i,j}^t) + L_{i+1,j+2^{n-1-i}}^t \\ R_{i+1,j}^t = r_{i+1,j}^{R,t} f(R_{i,j}^t, R_{i+1,j+2^{n-1-i}}^t + L_{i+1,j+2^{n-1-i}}^t) \\ R_{i+1,j+2^{n-1-i}}^t = r_{i+1,j+2^{n-1-i}}^{R,t} f(R_{i,j}^t, L_{i+1,j}^t) + R_{i,j+2^{n-1-i}}^t \end{cases} \quad (5)$$

WBP 译码算法的神经网络架构基于 T 次迭代，每次迭代包含左向传播和右向传播，所以一个完整的 WBP 译码算法的神经网络结构包含 1 层输入层， $2(n-1)T+1$ 层隐藏层， 1 层输出层，如图 5 所示。

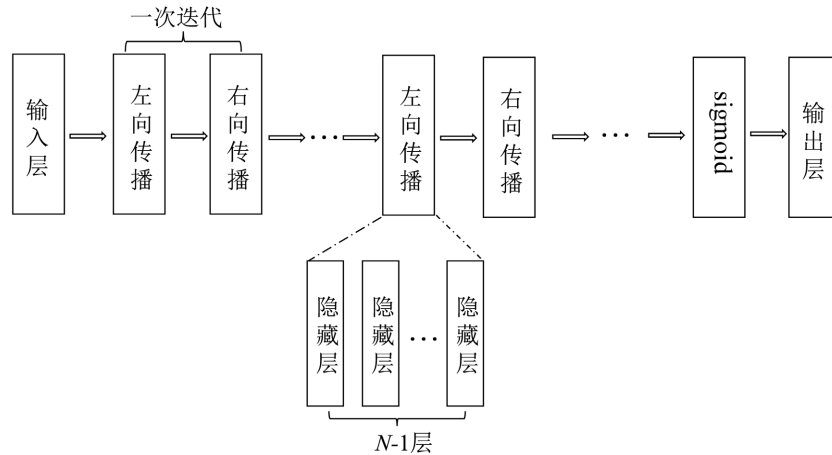


Figure 5. The BP decoding factor plot is transformed into an NN network with weights
图 5. BP 译码因子图转化为带有权重的 NN 网络

最后一层的激活函数采用 sigmoid 函数, 用于在 BP 译码结束时将 LLR 值映射为(0, 1), 可以看作是硬判决, 便于译码器利用监督学习进行训练。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

虽然更深层次的神经网络通常具有更大的解码能力, 但也带来巨大的加权参数计算。所以, 为了在误码率性能和复杂度之间保持良好的平衡, 迭代次数的选择需要适当。与传统的 BP 算法相比, 文献[19]在深度学习框架 Tensorflow 上, 设置学习率 0.001, 采用 Adam 自适应优化器, 训练加权参数。训练集经过 AWGN 信道传输后均为零码字。信噪比(SNRs)范围为 1~4 dB。共生成 24,000 个码字, 批次大小为 120 (每个信噪比 30 个码字)。该模型采用早停机制来提前结束译码。权值初始化是神经网络训练的关键部分, 为了避免局部优化, 权重参数初始化为 1。测试集由随机二进制消息(每个信噪比 10,000 帧)经过编码、二进制相移键控(BPSK)调制并通过 AWGN 信道传输后生成。训练神经网络的目的是最小化损失函数并训练出最优的权重参数。仿真结果如图 6 所示, 采用 T = 5 的 WBP 译码算法的译码性能优于 T = 30 的传统 BP 译码算法, 可以实现更低的误码率。

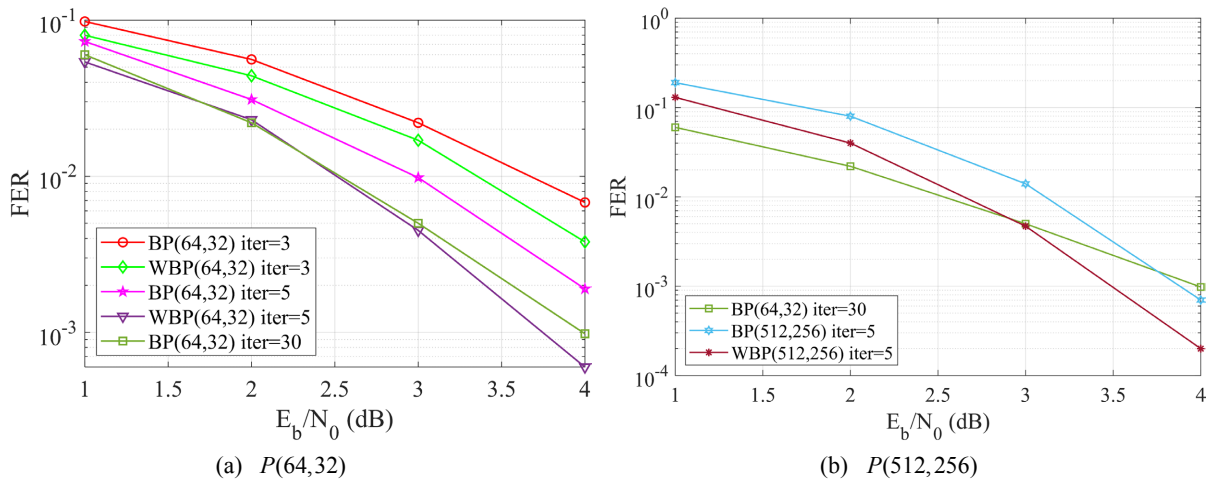


Figure 6. Performance comparison between WBP and BP decoding algorithms
图 6. WBP 与 BP 译码算法性能对比图

3.2. 集成学习模型

3.2.1. 弱学习器及数据集

弱学习器通常是用一个现有的学习算法由训练数据训练产生，例如决策树算法、神经网络算法等。根据弱学习器所采用的学习算法是否相同可分为同质集成或异质集成。同质集成中，弱学习器由相同的学习算法生成，异质则由不同算法生成。鉴于神经网络模型的 WBP 译码算法在译码性能和时延两方面均优于传统 BP 算法，所以弱学习器选择基于神经网络的 WBP 译码器。为了实现集成学习性能优于弱学习器的目标，要求弱学习器要相互独立。对于同质集成来说，则需要利用不同的参数构成具有不同译码性能的译码器，所以需要独立的数据集来保证权重参数的不同。

WBP 译码器的权重参数训练数据集一般采用全零码字，这种数据集包含的信息只有码长 N 及信息位 K ，由于无法区分，所以不能作为同质集成的训练数据集。为了保证数据集的独立，首先分析极化码信息传输过程。假设传输信道为加性高斯白噪声(AWGN)信道，采用二进制相移键控(BPSK)调制。信源序列 u 经过极化码编码后，由信道 W_N 传输，译码器接收信息序列为 y ：

$$y = (1 + 2x) + w \quad (7)$$

式中 w 为 AWGN 信道噪声，服从 $N(0, \sigma^2)$ 分布。

信息序列 y 对应的 LLR 值为 LLR_i ：

$$LLR_i = \ln \frac{W(y_i | 0)}{W(y_i | 1)} = \ln \frac{\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - 1)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i + 1)^2}{2\sigma^2}\right)} = \frac{2y_i}{\sigma^2} \quad (8)$$

再由 LLR 值为 LLR_i 进行硬判决，得到 \hat{u} ， \hat{u} 为未经 BP 译码的信源序列估计值。定义信道输出的二进制误比特分布 $P(e)$ ，其中 $e = \hat{u} \oplus c$ ， c 为传输码字，也就是全零序列。而由于信道不同，不同信道传输后的 \hat{u} 有很大差别，所以可以以此来采样数据。二进制误比特集合为 $\varepsilon = \{e^{(1)}, e^{(2)}, \dots, e^{(k)}\}$ ，按照一定规律进行分类训练。定义二进制比特集合的不同误差分布：

$$\varepsilon = \bigcup_{i=1}^{\alpha} \chi^{(i)} : \chi^{(i)} \cap \chi^{(j)} = \emptyset, \forall i \neq j \quad (9)$$

训练数据集 $D = \{D^{(1)}, D^{(2)}, \dots, D^{(\alpha)}\}$ ，每个 BP 译码器都有一个独立的数据集：

$$D^i = \{\chi^{(i)} : e^{(k)} \in \chi^{(i)}\} \quad i \in \{1, 2, \dots, \alpha\} \quad (10)$$

汉明距离主要应用在通信编码领域上，常用于制定可纠错的编码体系。在机器学习领域中，汉明距离是一种常用的距离度量方式。当传输码字 c 为全零码字时，根据译码器接收到的未经译码的信源估计值 \hat{u} 计算出非 0 比特的个数，作为二进制错误比特数据区分的依据：

$$\chi^{(i)} = \{e^{(k)}\} \quad k \text{ 为包含非 0 比特数} \quad (11)$$

3.2.2. 强学习器

强学习器是通过一定的方式对弱学习器集成，达到超越弱学习器性能的组合体。集成学习的结合策略主要有平均法、投票法和学习法。平均法是将各个弱学习器的输出进行平均或加权平均，作为集成学习的结果。投票法则是对各个弱学习器的输出投票打分，将投票结果作为集成学习的输出。学习法是将所训练出的弱学习器的学习结果作为输入，重新训练一个更好的学习器来得到最终结果。经过独立数据集训练出具有不同权重参数的 WBP 译码器相比于传统 BP 译码算法已经具有很大的译码优势，同时为了

符合译码的特点，采用了类似 BPL 译码器的组合策略。

BPL 译码算法基于不同排列的因子图来实现 L 个并行的 BP 译码器译码，除了随机选择，列表的选择可以基于多种方案，如最小欧式距离路径选择、和积路径选择、基于码字权重的路径选择等方案。这些选择方案是为了从多个 BP 译码器中筛选出译码性能最好的结果输出。但其并不能将 BPL 译码器看作是 BP 译码器的集成，因为 BPL 译码器提供的更多是一种广度搜索的可能，并没有集成为强学习器的功能。显然，集成学习是通过构建并结合多个弱学习器来完成学习任务，多个经过训练的弱学习器集成起来就是一个强学习器。更重要的是，BPL 译码器的输入是相同的，利用一定规则来决定哪条路径更可能为正确译码。所以基于 BPL 译码器的思想，新提出一种学习器筛选机制作为结合策略，这种结合策略的目的是得到一个广度的搜索空间，以便于所提出的译码器可以适应更复杂的信道环境。

极化码译码的过程，受到信道环境的变化，理论上不存在绝对完美的译码器，译码器的译码结果只是译码的一种概率。为了获得更好的译码性能，利用学习器筛选机制从这 α 个学习器里选出译码性能更好的 β 个学习器，并将这些由不同权重参数的 WBP 译码器组合而成的学习器，称为强学习器，即 E-WBP 译码器。

所以 E-WBP 译码器组成不同于常用的集成学习结合策略，而是在广度上增加译码成功的选择，提高更多的译码可能。学习器的筛选机制可以借助 BPL 译码算法路径选择中一种常见简单的算法，即最小路径选择算法来筛选。测试集由随机二进制消息经过编码、二进制相移键控(BPSK)调制并通过 AWGN 信道传输后生成。定义 \mathbf{u} 为测试集传输码字， $\mathbf{u} = \{u_1, u_2, \dots, u_N\}$ ，定义 \mathbf{c} 为码字经过 BPSK 调制 ($0 \rightarrow 1, 1 \rightarrow -1$)，然后经 AWGN 信道传输，则接收到的信息序列 $\mathbf{y} = \mathbf{c} + \mathbf{n}$ ，其中 $\mathbf{n} \sim N(0, \sigma^2)$ ， $\mathbf{y}_1^N = \{y_1, y_2, \dots, y_N\}$ 。计算欧式距离：

$$d_E^2(\mathbf{y}, \mathbf{c}) = \sum_{i=1}^N (y_i - c_i)^2 \quad (12)$$

将上式展开：

$$d_E^2(\mathbf{y}, \mathbf{c}) = \sum_{i=1}^N y_i^2 - 2 \sum_{i=1}^N y_i \cdot c_i + \sum_{i=1}^N c_i^2 \quad (13)$$

利用测试集，计算最小欧式距离，从 α 个学习器里筛选出 β 个学习器组成 E-WBP 译码器。

3.3. E-WBP 译码算法流程

基于集成学习的 BP 译码器，该译码器集成了 WBP 译码算法和 BPL 译码算法的优势。E-WBP 译码算法流程如图 7 所示。

独立数据集：训练数据集采用全零码字，为了形成独立的数据集，对信道传输后得到的信源序列 \mathbf{y} 进行硬判决，从而得到未经译码的码字估计 $\hat{\mathbf{u}}$ ，计算出二进制错误比特，再根据二进制错误比特的分布进行分区。为方便算法实现，采用简单的汉明距离来区分数据集。从而得到利用汉明距离分类器对训练数据集进行采集，形成不同的子数据集 $D = \{D^{(1)}, D^{(2)}, \dots, D^{(\alpha)}\}$ 。

WBP 译码器参数训练：由不同的子数据集独立训练出 WBP 译码器 $F_1^{D^{(1)}, \omega^{(1)}}, F_2^{D^{(2)}, \omega^{(2)}}, \dots, F_\alpha^{D^{(\alpha)}, \omega^{(\alpha)}}$ ， $D^{(\alpha)}$ 表示第 α 个子数据集， $\omega^{(\alpha)}$ 为训练的参数，神经网络模型采用 DNN 网络[19]，共训练出 α 个弱学习器，网络中采用的二进制交叉熵损失函数来衡量 WBP 译码器的输入值和期望值的差距，并通过两者的差距来优化译码器的权重参数，得到译码性能更优异的译码器，其中输入值和期望值都是全零码字。权重参数的训练的最终目的就是最小化损失函数并得到最优的权重参数。二进制交叉熵损失函数 $L(u, \hat{u})$ ：

$$L(u, \hat{u}) = -\frac{1}{N} \sum_{i=1}^N u_i \log_2 \hat{u}_i + (1 - u_i) \log_2 (1 - \hat{u}_i) \quad (14)$$

学习器筛选: 由随机二进制消息生成测试集, 并计算弱学习器译码后的最小欧式距离, 筛选出 β 个弱学习器 $\{F_1^{D^{(1)}, \omega^{(1)}}, F_2^{D^{(2)}, \omega^{(2)}}, \dots, F_\beta^{D^{(\beta)}, \omega^{(\beta)}}\}$ 组成 E-WBP 译码器。组成 E-WBP 译码器的权重参数 $\{\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(\beta)}\}$ 都可以进行离线训练。

E-WBP 译码: E-WBP 译码器接收到的信号序列 y_1^N , 离线训练得到的权重参数传至 BP 译码器, 形成 β 个 WBP 译码器, 同时也是集成后的强学习器(E-WBP 译码器)。接收到相同的接收信号后, 译码得到 β 个估计序列 \hat{u} , 再利用生成矩阵的早停机制提前终止译码, 最后再利用欧式距离判决来对候选码字进行判决输出最后的译码码字 \hat{x}_{BPL} 。

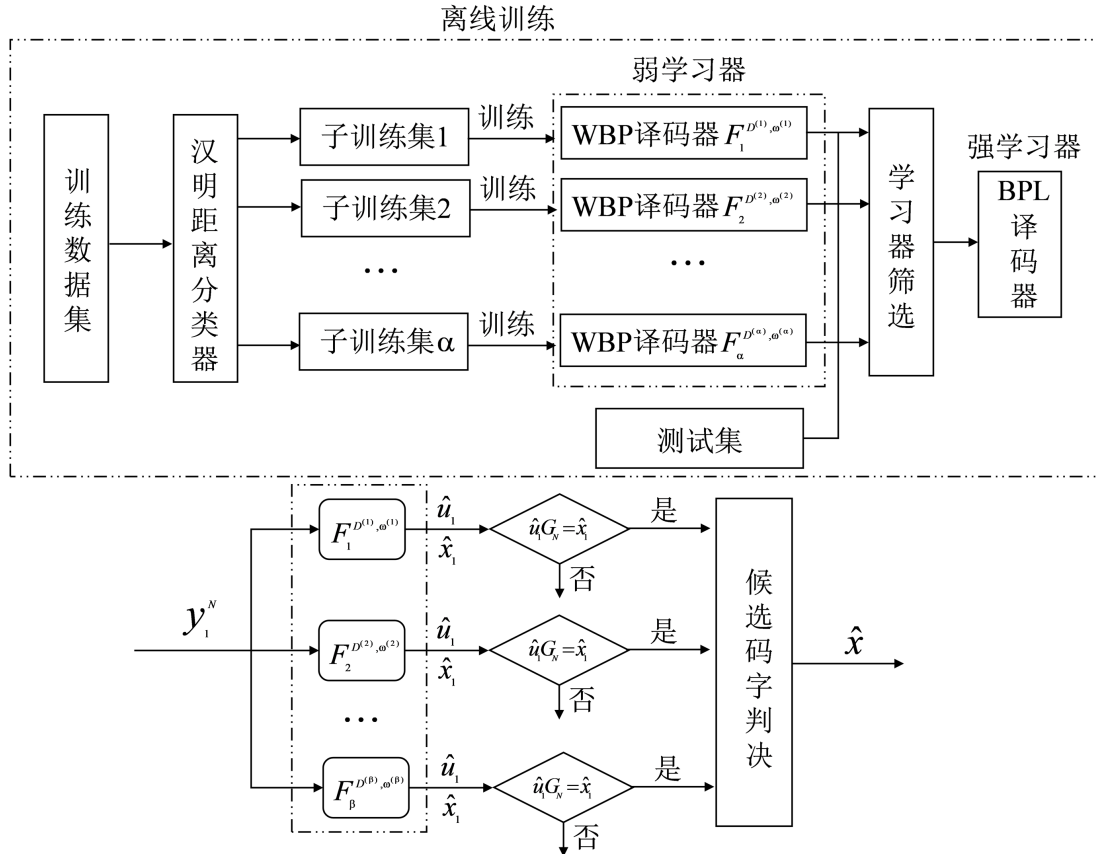


Figure 7. Schematic diagram of E-WBP decoding algorithm

图 7. E-WBP 译码算法流程示意图

4. 仿真结果与分析

由于弱学习器 $F_i^{D^{(i)}, \omega^{(i)}}$ 可以采用离线训练形式, 其训练过程的计算复杂度可以忽略。在训练过程中, 每个 WBP 译码器最多迭代 T 次, 每个最小计算单元会含有 4 个参数, 每个弱译码器训练出 $4T \log N$ 参数, N 为码长, 一般码长固定为 $\{64, 128, 256, 512, 1024\}$, E-WBP 译码器需要离线训练 $4\beta \cdot T \log N$ 权重参数。学习器的筛选只是对训练出来的弱学习器进行评价, 以便组合成强学习器, 这里忽略其带来的复杂度。另一部分复杂度主要来自 E-WBP 译码过程中, E-WBP 译码与 BPL 译码的过程相似。而 BPL 译码器的计算复杂度是 $O(\beta N \log_2 N)$, 同时极化码采用生成矩阵的早停机制, 需要在每次迭代后检查估计的信源序列与码字估计的生成关系, 是否满足 $\hat{x} = \hat{u}G_N$, 如果满足条件, 停止译码, 其复杂度为 $O(N \log_2 N)$ 。所以 E-WBP 译码器在译码过程中的计算复杂度为 $O((\beta + 1)N \log_2 N)$, 空间复杂度为

$O(LN)$ 。

如图 7 所示, E-WBP 译码器由离线训练产生, 主要包括权重参数的训练和测试。训练样本设置传输码字为全零码字, 经过 BPSK 调制为全 1 序列, 传输信道为 AWGN 信道, 再根据接收到的信号 y_1^N 计算出对数似然比 $LLR_i = \{L_1, L_2, \dots, L_N\}$, 得到未经译码的全零码字估计值 \hat{u} , 再由汉明距离区分为 α 个独立的数据集。训练阶段中 α 个 WBP 译码器每个训练批次包括 10000 个样本, 每个样本由特定的信噪比 $\{2, 3, 4, 5, 6\}$ dB 等概率生成。测试数据由随机信息序列编码生成, 采用高斯近似构造, 经 BPSK 调制后在 AWGN 信道上传输。训练好的弱学习器再经过测试集, 再利用学习器筛选机制筛选出 β 个 WBP 译码器用于组成 E-WBP 译码器。设置 $\alpha = 8$, $\beta = 4$, 组成 E-WBP 译码器。使用 PyTorch 框架搭建神经网络, 使用 NVIDIA GeForce RTX 3090 显卡加速训练。弱学习器的训练参数如表 1:

Table 1. WBP decoder training parameters

表 1. WBP 译码器训练参数

设置	参数
学习框架	PyTorch
优化器	Adam
学习率	0.001
样本批次	10,000
样本信噪比	$\{2, 3, 4, 5, 6\}$ dB
损失函数	二进制交叉熵损失函数
码字随机数	345,000
噪音随机数	786,000
码长	$N = \{256, 512\}$
码率	$K = 1/2$
BP 迭代次数	$T = 6$

为了评估本文所提算法的译码性能, 对该算法在不同码长情况下进行仿真模拟。首先仿真码长为 $N = 256$, $K = 128$ 的 BP、BPL [5]、WBP [19] 和 E-WBP 译码算法的 FER 性能。迭代次数均为 6 次, BPL 译码算法的 L 为 4, 与 E-WBP 译码算法的 $\beta = 4$ 相匹配。

由图 8 可以看出, 在相同迭代次数时, 传统的 BP 译码器 FER 性能弱于经过神经网络训练得到的 WBP 译码器, WBP 译码器稍弱于 $L = 4$ 的 BPL 译码器译码性能, 但优势在于 WBP 占用更少的译码空间。E-WBP 则具有接近于 BPL 的译码性能。图 9 则是对码长 $N = 526$, $K = 256$ 的极化码的 4 种译码算法对比, 可以看出 E-WBP 译码算法已经取得了与 BPL 相同的译码性能, 同时相较于 $FER = 10^{-2}$ 时, 译码性能具有 0.3 Db 的增益。另一方面, 传统的 BP 译码算法和 BPL 译码算法, 消耗的计算资源多, 时延较长, 而 E-WBP 译码算法离线训练所需要的计算复杂度和存储空间占用少, 以更少的迭代次数取得更好的译码性能, 更适合部署在实际通信系统中。

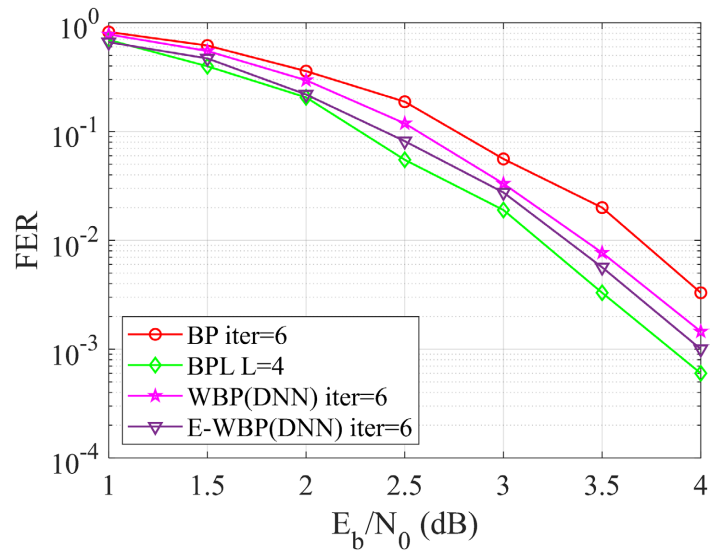


Figure 8. Comparison of FER performance of four decoder algorithms for polarized codes (256, 128)
图 8. 极化码(256, 128)的 4 种译码器算法 FER 性能对比

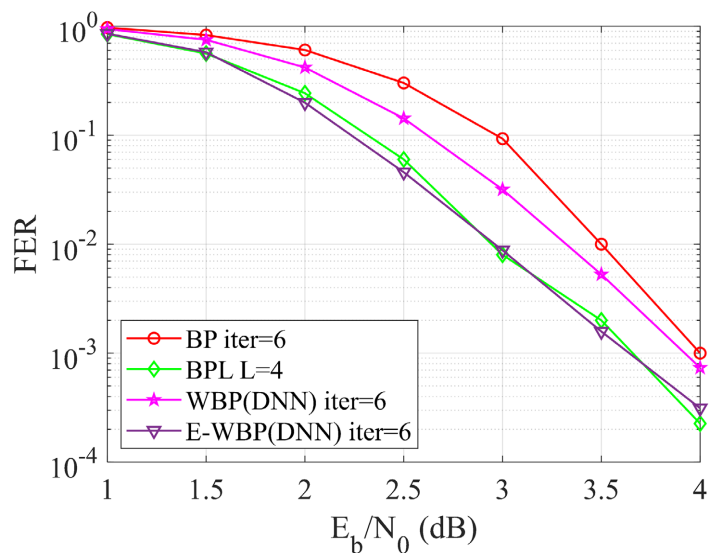


Figure 9. Comparison of FER performance of four decoder algorithms for polarized codes (512, 256)
图 9. 极化码(512, 256)的 4 种译码器算法 FER 性能对比

5. 结束语

本文提出一种基于集成学习的极化码 BP 译码算法, 该算法通过集成 WBP 译码来构建强学习器形成新的译码器, 达到由于 WBP 译码器的目的。同时, 该算法利用离线训练, 并且可以实现参数共享可以有效地减少内存开销和迭代次数。仿真结果表明, 该算法在译码性能和复杂度之间取得良好的平衡, 取得了与 BPL 译码器接近的译码性能。

参考文献

- [1] 崔晨. 不等差错保护极化码编码方法研究[D]: [博士学位论文]. 哈尔滨: 哈尔滨工业大学, 2022.
- [2] 张川, 申怡飞, 周文岳, 季厚任. 5G 信道编译码: 算法与实现[M]. 北京: 科学出版社, 2021.

- [3] Simsek, C. and Turk, K. (2016) Simplified Early Stopping Criterion for Belief-Propagation Polar Code Decoders. *IEEE Communications Letters*, **20**, 1515-1518. <https://doi.org/10.1109/LCOMM.2016.2580514>
- [4] Elkelesh, A., Ebada, M., Cammerer, S., et al. (2018) Belief Propagation List Decoding of Polar Codes. *IEEE Communications Letters*, **22**, 1536-1539. <https://doi.org/10.1109/LCOMM.2018.2850772>
- [5] Doan, N., Hashemi, S.A., Mondelli, M., et al. (2018) On the Decoding of Polar Codes on Permuted Factor Graphs. 2018 *IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, 9-13 December 2018, 1-6. <https://doi.org/10.1109/GLOCOM.2018.8647308>
- [6] Elkelesh, A., Ebada, M., Cammerer, S., et al. (2018) Belief Propagation Decoding of Polar Codes on Permuted Factor Graphs. 2018 *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, 15-18 April 2018, 1-6. <https://doi.org/10.1109/WCNC.2018.8377158>
- [7] Arlı, A.Ç. and Gazi, O. (2019) Noise-Aided Belief Propagation List Decoding of Polar Codes. *IEEE Communications Letters*, **23**, 1285-1288. <https://doi.org/10.1109/LCOMM.2019.2918535>
- [8] Geiselhart, M., Elkelesh, A., Ebada, M., et al. (2020) CRC-Aided Belief Propagation List Decoding of Polar Codes. 2020 *IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, 21-26 June 2020, 395-400. <https://doi.org/10.1109/ISIT44484.2020.9174249>
- [9] Ren, Y., Shen, Y., Zhang, Z., et al. (2020) Efficient Belief Propagation Polar Decoder with Loop Simplification Based Factor Graphs. *IEEE Transactions on Vehicular Technology*, **69**, 5657-5660. <https://doi.org/10.1109/TVT.2020.2979334>
- [10] Yu, Y., Pan, Z., Liu, N., et al. (2019) Belief Propagation Bit-Flip Decoder for Polar Codes. *IEEE Access*, **7**, 10937-10946. <https://doi.org/10.1109/ACCESS.2019.2891951>
- [11] Shen, Y., Song, W., Ren, Y., et al. (2020) Enhanced Belief Propagation Decoder for 5G Polar Codes with Bit-Flipping. *IEEE Transactions on Circuits and Systems II: Express Briefs*, **67**, 901-905. <https://doi.org/10.1109/TCSII.2020.2984536>
- [12] Shen, Y., Song, W., Ji, H., et al. (2020) Improved Belief Propagation Polar Decoders with Bit-Flipping Algorithms. *IEEE Transactions on Communications*, **68**, 6699-6713. <https://doi.org/10.1109/TCOMM.2020.3017656>
- [13] Doan, N., Hashemi, S.A. and Gross, W.J. (2020) Decoding Polar Codes with Reinforcement Learning. *GLOBECOM 2020-2020 IEEE Global Communications Conference*, Taipei, 7-11 December 2020, 1-6. <https://doi.org/10.1109/GLOBECOM42002.2020.9348007>
- [14] 徐鹏, 孔令军, 赵生妹, 郑宝玉. 基于 PD-CNN 的 Polar 码译码算法[J]. 信号处理, 2019, 35(10): 1652-1660.
- [15] 何彦琦, 彭大芹, 赵雪志. 一种基于循环神经网络的极化码 BP 译码算法[J]. 计算机工程, 2022, 48(1): 197-203.
- [16] Nachmani, E., Be'ery, Y. and Burshtein, D. (2016) Learning to Decode Linear Codes Using Deep Learning. 2016 *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, 27-30 September 2016, 341-346. <https://doi.org/10.1109/ALLERTON.2016.7852251>
- [17] Nachmani, E., Marciano, E., Lugosch, L., et al. (2018) Deep Learning Methods for Improved Decoding of Linear Codes. *IEEE Journal of Selected Topics in Signal Processing*, **12**, 119-131. <https://doi.org/10.1109/JSTSP.2017.2788405>
- [18] Lugosch, L. and Gross, W.J. (2017) Neural Offset Min-Sum Decoding. 2017 *IEEE International Symposium on Information Theory (ISIT)*, Aachen, 25-30 June 2017, 1361-1365. <https://doi.org/10.1109/ISIT.2017.8006751>
- [19] Xu, W., Wu, Z., Ueng, Y.L., et al. (2017) Improved Polar Decoder Based on Deep Learning. 2017 *IEEE International Workshop on Signal Processing Systems (SiPS)*, Lorient, 3-5 October 2017, 1-6. <https://doi.org/10.1109/SiPS.2017.8109997>
- [20] Teng, C.F., Wu, C.H.D., Ho, A.K.S., et al. (2019) Low-Complexity Recurrent Neural Network-Based Polar Decoder with Weight Quantization Mechanism. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, 12-17 May 2019, 1413-1417. <https://doi.org/10.1109/ICASSP.2019.8683778>
- [21] Huang, Y., Zhang, M. and Dou, Y. (2020) A Low-Complexity Residual Neural Network Based BP Decoder for Polar Codes. 2020 *International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, 21-23 October 2020, 889-893. <https://doi.org/10.1109/WCSP49889.2020.9299762>
- [22] Oliveira, L.M., Oliveira, R.M. and de Lamare, R.C. (2021) Decoding of Polar Codes Based on Q-Learning-Driven Belief Propagation. <https://arxiv.org/abs/2111.00673>
- [23] Raviv, T., Raviv, N. and Be'ery, Y. (2020) Data-Driven Ensembles for Deep and Hard-Decision Hybrid Decoding. 2020 *IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, 21-26 June 2020, 321-326. <https://doi.org/10.1109/ISIT44484.2020.9173971>
- [24] Raviv, T., Goldmann, A., Vayner, O., et al. (2023) CRC-Aided Learned Ensembles of Belief-Propagation Polar Decoders. <https://arxiv.org/abs/2301.06060>